

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims:

1-5 (Cancelled)

6. **(Currently Amended)** A method as recited in claim ~~[[4]]~~ **16**, wherein said method further comprises:

not providing a data entry in the data stream for the **first** command when the **first** command does not have data associated with it.

7. **(Currently Amended)** A method as recited in claim ~~[[6]]~~ **16**, wherein said method further comprises:

writing a representation of a second command associated with another instruction into a second code entry of the code stream;

determining whether the second command has data associated with it; and

writing a representation of the associated data or a reference to a representation of the data associated with the second command into a second data entry of the data stream when the command has associated data.

8-9 (Cancelled)

10. **(Currently Amended)** A method as recited in claim ~~[[9]]~~ **7**, wherein the first and second entries of the code stream are adjacent to each other.

11-15 (Cancelled)

16. **(Currently Amended)** A method of executing computer instructions ~~[[on]]~~ **by** a virtual machine **that includes an interpreter**, the method comprising:

receiving a single stream of Bytecode instructions in a class file for execution by the virtual machine;

converting, at load time prior to execution time, the single stream of Bytecode instructions into a pair of streams in the virtual machine, wherein the pair of streams includes a code stream and a data stream, wherein the code stream includes only virtual machine commands, wherein the data stream includes only data associated with virtual machine commands stored in the code stream, and wherein said converting further comprises:

determining whether at least one Bytecode which represents a first command in the single stream of Bytecodes has associated data in the single stream;

accessing a constant pool associated with the class file when said determining determines that the first command in said single stream has data associated with it;

determining, based on the accessing of the constant pool, at least one data parameter value for the first command;

loading, prior to execution time, the first command in said code stream in the virtual machine;

loading, prior to execution time, the at least one data parameter value in the data stream in the virtual machine;

fetching, by the interpreter at execution time, [[a]] the first command associated with a virtual machine computer instruction from [[a]] the code stream in the virtual machine;

determining, by the interpreter at execution time, whether [[the]] the first command has an associated data parameter value in the data stream;

fetching, by the interpreter at execution time, from [[a]] the data stream in the virtual machine the associated at least one data parameter value of the command when said determining determines that the first command has an associated data parameter value; and

executing by said interpreter at execution time the first command with the at least one data the associated parameter value [[s]] when said determining determines that the first command has an associated data parameter value after the associated parameter of the commands have been fetched thereby executing the first command without processing the constant pool at execution time in order to determine the at least one data parameter value needed to execute the first command.

17. (Original) A method as recited in claim 16, wherein the method further comprises:

updating a pointer to the command stream; and
updating a pointer to the data stream.

18. (Currently Amended) A method as recited in claim 16, wherein the code stream includes ~~Java™-compliant~~ commands represented as bytecodes, and the ~~code~~ data stream includes ~~data associated with the Java™-commands~~ represented as bytecodes.

19. (Cancelled)

20. (Currently Amended) A method as recited in claim ~~19~~ 16, wherein the Java™ first command[[s]] can be a load constant command, an invoke method command, a jump command, an instantiation command, or a get/put field command.

21. (Cancelled)

22. (Currently Amended) A method as recited in claim ~~21~~ 16, wherein said converting of said stream further comprises:

writing a representation of [[a]] the first command associated with a first instruction into a code entry of the code stream;

determining whether the first command has associated data; and

reading the associated data from a Constant Pool when the command has ~~an~~ associated data;

processing the associated data from the Constant Pool ~~the associated data from the Constant Pool~~;

writing a representation of the associated data or a reference to a representation of the data associated into the Java™ code stream after said processing of the associated data;

~~wherein each Java™-command and data associated with that Java™-command are represented respectively in one or more bytecodes of the Java™-code stream and the Java™-data stream, and~~

~~wherein each bytecode can be one or more bytes.~~

23. **(Currently Amended)** A method as recited in claim 22, wherein said processing operates to determine a constant value associated with a Java[™]-Load Constant command.

24. **(Currently Amended)** A method as recited in claim 22, wherein said processing operates to determine a reference to a method invocation cell that includes information relating to a Java[™]-invoke method command.

25. **(Currently Amended)** A method as recited in claim 22, wherein said processing ~~operates to determine the code stream offset and data stream offset associated with a~~ Java[™] Jump command.

26. **(Currently Amended)** A method as recited in claim 22, wherein said processing operates to process a Constant Pool associated with an Java[™]-instantiation command.

27. **(Currently Amended)** A method as recited in claim 22, wherein said processing operates to process a Constant Pool associated with a Java[™]-Get/Put field command.

28-30 (Canceled)

31. **(New)** A computer system for executing computer instructions by a virtual machine that includes an interpreter, wherein said computer system is capable of:
receiving a single stream of Bytecode instructions in a class file for execution by the virtual machine;

converting, at load time prior to execution time, the single stream of Bytecode instructions into a pair of streams in the virtual machine, wherein the pair of streams includes a code stream and a data stream, wherein the code stream includes only virtual machine commands, wherein the data stream includes only data associated with virtual machine commands stored in the code stream, and wherein said converting further comprises:

determining whether at least one Bytecode which represents a first command in the single stream of Bytecodes has associated data in the single stream;

accessing a constant pool associated with the class file when said determining determines that the first command in said single stream has data associated with it;

determining, based on the accessing of the constant pool, at least one data parameter value for the first command;

loading, prior to execution time, the first command in said code stream in the virtual machine;

loading, prior to execution time, the at least one data parameter value in the data stream in the virtual machine;

fetching, by the interpreter at execution time, the first command from the code stream in the virtual machine;

determining, by the interpreter at execution time, whether the first command has an associated data parameter value in the data stream;

fetching, by the interpreter at execution time, from the data stream in the virtual machine the associated at least one data parameter value when said determining determines that the first command has an associated data parameter value; and

executing by said interpreter at execution time the first command with the at least one data parameter value when said determining determines that the first command has an associated data parameter value, thereby executing the first command without processing the constant pool at execution time in order to determine the at least one data parameter value needed to execute the first command.

32. (New) A computer system as recited in claim 31, wherein said computer system is further capable of:

not providing a data entry in the data stream for the first command when the first command does not have data associated with it.

33. (New) A computer system as recited in claim 31, wherein said computer system is further capable of:

writing a representation of a second command associated with another instruction into a second code entry of the code stream;

determining whether the second command has data associated with it; and

writing a representation of the associated data or a reference to a representation of the data associated with the second command into a second data entry of the data stream when the command has associated data.

34. (New) A computer system as recited in claim 33, wherein the first and second entries of the code stream are adjacent to each other.

35. (New) A computer system as recited in claim 31, wherein said computer system is further capable of:

- updating a pointer to the command stream; and
- updating a pointer to the data stream.

36. (New) A computer system as recited in claim 33, wherein the first command can be a load constant command, an invoke method command, a jump command, an instantiation command, or a get/put field command.

37. (New) A computer system as recited in claim 31, wherein said converting of said stream further comprises:

- writing a representation of the first command associated with a first instruction into a code entry of the code stream;
- determining whether the first command has associated data;
- reading the associated data from a Constant Pool when the command has associated data;
- processing the associated data from the Constant Pool; and
- writing a representation of the associated data or a reference to a representation of the data associated into the code stream after said processing of the associated data.

38. (New) A computer readable medium including computer program code for executing computer instructions by a virtual machine that includes an interpreter, wherein said computer readable medium includes:

- program code for receiving a single stream of Bytecode instructions in a class file for execution by the virtual machine;

- program code for converting, at load time prior to execution time, the single stream of Bytecode instructions into a pair of streams in the virtual machine, wherein

the pair of streams includes a code stream and a data stream, wherein the code stream includes only virtual machine commands, wherein the data stream includes only data associated with virtual machine commands stored in the code stream, and wherein said converting further comprises:

- determining whether at least one Bytecode which represents a first command in the single stream of Bytecodes has associated data in the single stream;

- accessing a constant pool associated with the class file when said determining determines that the first command in said single stream has data associated with it;

- determining, based on the accessing of the constant pool, at least one data parameter value for the first command;

- program code for loading, prior to execution time, the first command in said code stream in the virtual machine;

- program code for loading, prior to execution time, the at least one data parameter value in the data stream in the virtual machine;

- program code for fetching, by the interpreter at execution time, the first command from the code stream in the virtual machine;

- program code for determining, by the interpreter at execution time, whether the first command has an associated data parameter value in the data stream;

- program code for fetching, by the Interpreter at execution time, from the data stream in the virtual machine the associated at least one data parameter value when said determining determines that the first command has an associated data parameter value; and

- program code for executing by said interpreter at execution time the first command with the at least one data parameter value when said determining determines that the first command has an associated data parameter value, thereby executing the first command without processing the constant pool at execution time in order to determine the at least one data parameter value needed to execute the first command.

39. (New) A computer readable medium as recited in claim 38, further comprising: computer program code for not providing a data entry in the data stream for the first command when the first command does not have data associated with it.

40. (New) A computer readable medium as recited in claim 38, further comprising:

computer program code for writing a representation of a second command associated with another instruction into a second code entry of the code stream;

computer program code for determining whether the second command has data associated with it; and

computer program code for writing a representation of the associated data or a reference to a representation of the data associated with the second command into a second data entry of the data stream when the command has associated data.

41. (New) A computer readable medium as recited in claim 38, wherein the first and second entries of the code stream are adjacent to each other.

42. (New) A computer readable medium as recited in claim 38, further comprising:
computer program code for updating a pointer to the command stream; and
computer program code for updating a pointer to the data stream.

43. (New) A computer readable medium as recited in claim 38, wherein the first command can be a load constant command, an invoke method command, a jump command, an instantiation command, or a get/put field command.

44. (New) A computer readable medium as recited in claim 38, wherein said computer code for converting of said stream further comprises:

computer program code for writing a representation of the first command associated with a first instruction into a code entry of the code stream;

computer program code for determining whether the first command has associated data;

computer program code for reading the associated data from a Constant Pool when the command has an associated data;

computer program code for processing the associated data from the Constant Pool; and

computer program code for writing a representation of the associated data or a reference to a representation of the data associated into the code stream after said processing of the associated data.